

TP1 IHM

Q1. Ils servent a mettre en pause / continuer, ou stopper le programme

Q2. Avance "pas a pas" : chaque déclaration, initialisation, tour de boucle, avance normale : ligne par ligne, saute les boucles, avance jusqu'à sortie : avance jusqu'au prochain bloc de code.

Q3. Le lien avec l'évènement paint est créé dans la méthode InitializeComponent de FeuilleDessin

Q4. "sender" est la source de l'évènement et "e" (de type PaintEventArgs) sont les paramètres de l'évènements

Q5.

- Pen : utilise un objet stylo pour dessiner
- Rectangle : créer un objet rectangle, avec les coordonnées du coin Haut Gauche et les dimensions, vide par défaut, les autres classes permettent de reellement dessiner
- Brushes : lié a Pen, créer un objet pinceau
- Point : créer un objet point défini pour intéragir avec
- Font : créer un objet de police pour écrire avec

Q6.

Refresh déclanche l'actualisation de la fenêtre, et donc l'animation

Q7.

Quand l'utilisateur appuie sur sa souris,

si "ellipse" et la souris sont superposés, mettre la variable texte à "Au revoir";

changer "ellipse" en Rectangle

actualise la fenêtre

Q8.

Comme elles vont êtres utilisées par plusieurs méthodes répondant à des évènements, on les déclare comme variables d'instance, et on les initialise a null et false dans le constructeur.

Q.9

x vaut le décalage entre l'abscisse de la souris au clic et l'abscisse actuelle

y vaut le décalage entre l'ordonnée de la souris au clic et l'ordonnée actuelle

current contient la position de l'ellipse

on modifie current pour le décaler selon x et y

l'ellipse se déplace au nouvel emplacement de current

départ prend la valeur de l'emplacement de la souris

on actualise la fenêtre

Q.10

Position = position du point haut gauche du noeud (rectangle)

Taille = objet Size qui contient une hauteur et une largeur, celles du noeud

méthode Dessine = avec Position et Taille, dessine un rectangle représentant le noeud sur la fenêtre

Q.11

Il doit être créé comme attribut d'instance de la fenêtre (FeuilleDessin)

Q.12

```
noeuds.ForEach(c => c.Dessine(e.Graphics));
```

pour chaque noeud "c" de la liste noeuds, appeler la méthode Dessine avec l'objet Graphics en paramètre

Q.13

A l'appui de la souris, créer un nouveau noeud aux coordonnées de la souris, de taille 10 par 10, et l'ajouter à la liste de noeuds, puis actualiser la fenêtre

Q.14

grâce au menu déroulant de la ToolStrip

Q.15

Source est le noeud de départ du trait

Destination est le noeud d'arrivée

Dessine est l'équivalent de Dessine pour un noeud, mais pour une ligne

Q.16

enDessinTrait est commun a plusieurs méthodes, il sera déclaré en haut de notre classe représentant la fenêtre

Q.17

Car on ne va récupérer le noeud de départ qu'a l'appui, et le noeud d'arrivée au relachement de la souris, sinon cela ferait trop d'opérations

